词向量聚类加权 TextRank 的关键词抽取*

夏 天
(中国人民大学数据工程与知识工程教育部重点实验室 北京 100872)
(中国人民大学信息资源管理学院 北京 100872)

摘要: 【目的】将维基百科蕴涵的世界知识以词向量方式融入 TextRank 模型, 改进单文档关键词抽取效果。【方法】利用 Word2Vec 模型基于维基百科中文数据, 生成词向量模型, 对 TextRank 词图节点的词向量进行聚类以调整簇内节点的投票重要性, 结合节点的覆盖和位置因素, 计算节点之间的随机跳转概率, 生成转移矩阵, 最终通过迭代计算获得节点的重要性得分, 选取前 TopN 个词语生成关键词。【结果】当 TopN≤7 时, 词向量聚类加权方法均优于对比方法; TopN=3 时, F 值取得最大值, 比先前最优结果增量提升了 3.374%; TopN>7 时, 结果与位置加权法相似。【局限】聚类分析使得计算开销变高。【结论】词向量聚类加权能够改善关键词抽取效果。

关键词: 关键词抽取 词向量 TextRank Word2Vec
分类号: G353

1 引 言

关键词抽取是指从给定的文本中自动抽取出若干有代表性的词语或词组, 用以反映文本的主要语义信息, 在图书情报领域有着广泛的应用。例如, 根据文献的关键词抽取结果构建词频矩阵, 在关键词级别上进行共词分析, 可以获取文献主题的发展变化, 进而支持图书馆海量数据的内容挖掘与分析。在实现策略方面, 关键词抽取既可以利用文本本身的内容和结构特征实现, 也可以通过对大量语料进行训练学习得到, 由于前者不需要先期训练过程, 实现相对简单, 并能达到令人满意的抽取效果, 因此近年来引起了广泛关注, 其中, TextRank 为该类算法的典型代表。

传统的 TextRank 算法仅利用了文档本身的信息, 如果能够将外部知识引入到关键词抽取过程之中, 理论上可以改善关键词抽取的效果, 2013 年以来兴起的词向量表示法, 能够将词语的语义投影到一个低维连续空间中, 并保持词语在语料库中的语义特点, 因此, 本文利用目前最大的维基百科在线开放知识库, 通过 Word2Vec 模型训练生成词向量, 并进行词向量聚类, 根据词语的聚类分布情况对 TextRank 词图节点进行非均匀加权, 从而将单一文档外部的世界知识融合进 TextRank 的计算过程中, 实现了具有较好效果的关键词抽取。

2 相关工作

TextRank 将链接分析中的 PageRank 算法思想引入到文本之中, 将特定粒度的文字单元及其共现关系表示为图结构, 并通过图的迭代计算实现重要性排序[1], 当以词语作为基本粒度时, 可以用于关键词抽取, 而以句子为基本粒度则可以用于文本摘要。由于其效果优于传统的 TF-IDF, 并且实现简单, 因此得到了广泛应用。

原始 TextRank 构建的词图中未考虑边的权重, 为进一步提高关键词抽取效果, 文献[2]将词语根据其位置加权, 从词语的覆盖影响力、位置影响力和频度影

通讯作者: 夏天, ORCID: 0000-0001-7564-7368, E-mail: xiat@ruc.edu.cn。

*本文系国家社会科学基金项目一般项目“我国数据新闻的理念、实践及其人才培养模式研究”(项目编号: 16BXW018)和北京高等学校青年英才计划项目“基于链接和主题分析的微博社区挖掘研究”(项目编号: YETP0215)的研究成果之一。

响力三个方面调整词图中边的传递权重，改进关键词抽取效果。文献[3]则进一步将 TextRank 与 LDA 主题模型融合到一起，综合考虑单一文档的结构信息和文档整体的主题信息，研究发现在数据集呈现明显的主题分布时，对关键词抽取效果有一定改善。文献[4]提出了 Tag-TextRank 方法，利用网页的社会化标签提高网页关键词抽取的效果。文献[5]在词语位置加权 TextRank 基础上，同时考虑词语的逆文档频率，实现关键词抽取并用于论文审稿自动推荐之中。近年来，随着 Word2Vec 词向量模型的兴起，人们开始尝试将 Word2Vec 应用于关键词抽取之中。文献[6]根据词向量之间的相似度进行词汇聚类，针对每个聚类结果簇选择距离质心最近的词语作为关键词，实现关键词抽取。文献[7]利用 Word2Vec 计算词汇之间的相似度矩阵，并融入到 TextRank 词图计算过程中，以改善抽取效果。

综上所述，在词图加权基础上，如何将文档外部信息纳入到 TextRank 的计算过程中，是改进 TextRank 关键词抽取的关键。已有的主题加权[3]、逆文档频度加权[5]等方法需要对待抽取文档本身所在的数据集进行预处理，结果因数据集不同而差异较大。Word2Vec 的训练数据独立于待抽取的文档，利用其训练生成的词向量对 TextRank 进行改进，理论上可以得到更为稳定的抽取结果。与文献[7]直接根据词向量相似度调整词语之间的跳转概率不同，本文首先对单一文档进行词向量聚类，进而根据词语与聚类质心的距离关系对词语重要性加权，构建新的概率转移矩阵，进行关键词抽取，并取得了最佳效果。

3　研究方法

基于 TextRank 的关键词抽取方法把关键词抽取问题转换为构成文档的词语的重要性排序问题，为此，笔者首先构建候选关键词词图(简称词图)，用于表示词语之间的结构关系；然后根据词语的词向量进行聚类分析，以词语在簇中的空间位置关系确定词语的聚类重要性，实现 TextRank 的聚类加权；最后构建完整的词语之间的概率转移矩阵，通过迭代运算获取节点的重要性，实现关键词排序和抽取。

3.1　词图构建

根据 TextRank 的基本思想，一篇文档可以根据其内部词语的邻接关系构成一个词图，进而根据词语在词图中的结构特征计算其重要性。为构建候选关键词图，基于前人研究[2]，将文本按句子分割，进行分词和词性标注，保留非单字词的名词、动词和形容词，构成词图的节点集 V，所有词语之间的邻接关系构成词图的边集 E，形成候选关键词图 $G=(V, E)$。在构建边时，假设词语 a 后面邻接出现词语 b，则同时在词图中增加两条有向边 $a \to b$ 和 $b \to a$，即词图 G 是一个有向图，如图 1 所示。

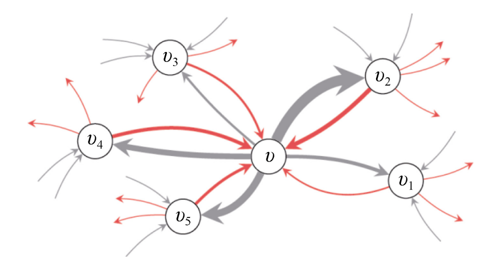


图 1　候选关键词词图示例

给定以上词图 $G=(V,E)$，令 $t(u)$ 表示节点 u 的 TextRank 值，则 $t(u)$ 可以采用公式(1)计算得到[1]。

$$t(u) = d\sum_{d \in adj[u]} p(v \to u)t(v) + (1-d)\frac{1}{|V|} \quad (1)$$

其中，$d \in [0,1]$ 为阻尼系数，表示任一节点均有 $1-d$ 的概率随机跳转到词图中的其他节点，以保证 TextRank 的迭代计算可以收敛，通常取值为 0.85；$adj[u] = \{v \mid (v \to u) \in E\}$ 表示节点 u 的相邻节点集；$p(v \to u)$ 表示由节点 v 到达 u 的随机跳转概率。

传统的 TextRank 算法在相邻节点之间采用均匀跳转策略，节点之间的跳转概率 $p(u \to v)$ 由公式(2)计算得到[1]。

$$p(u \to v) = \begin{cases} \dfrac{1}{\deg(u)}, & if\ \exists(u \to v) \in E \\ 0 & Otherwise \end{cases} \quad (2)$$

其中，$\deg(u)$ 为节点 u 的度；以图 1 为例，节点 v 跳转到任一相邻节点的概率 $p(v \to v_i \mid i \in [1,\cdots,5]) = 0.2$。为改进 TextRank，文献[2]提出根据节点重要性进行非均匀跳转的优化策略，取得了较好效果，然而该研究在计算跳转概率时，仅利用文档本身有关的信息，即根据所跳转到词语节点的出现位置、频度进行加权。为利用文档外部信息优化跳转概率的赋值，笔者进一步提出词向量聚类加权算法。

3.2 词向量聚类加权

2013 年，Mikolov 等发布了词向量训练工具Word2Vec，利用浅层神经网络模型自动学习词语在语料库中的出现情况，把词语嵌入到一个维度适中的空间中，即 $words \to R^n$，维度 n 通常在 100 至 500 之间，词语在新空间 R^n 中的表示结果即为词向量[8]。相比于传统的文本表示法，Word2Vec 生成的词向量不仅维度较底，词语之间的语义和句法关联关系在空间中也能得到很好的体现，一方面语义相近的词语在空间中的距离也相近；另一方面，词向量之间的线性操作结果与人的理解也相符合。可以说，Word2Vec 训练学习得到的词向量蕴涵了词语在大规模数据集中的语义信息，因此，可以利用文档的词向量之间的关系，对 TextRank 词图节点之间的跳转概率进行加权。

本研究假设如下：词向量反映了世界整体信息，一篇文档可以根据词向量之间的相似度聚为若干簇，一个词语距离所在簇的质心越远，则越能反映一个簇的区别于质心附近词语的不同方面信息，在作为 TextRank 中的词语节点时，其投票的重要性越高，在与之相邻的节点之间拥有更高的跳转概率。

给定文档 d 及其包含的候选关键词词语集合 $\{w_1, w_2, \cdots, w_n\}$ 以及训练得到的 Word2Vec 词向量模型，令 $\overline{w_i}$ 表示词语 w_i 对应的词向量，令 $C = \{C_1, C_2, \cdots, C_k\}$ 表示由文档的词向量集合进行 K 均值聚类后的聚类结果，笔者提出公式(3)计算任一词语 u 在所隶属的簇 C_u 中的投票重要性。

$$VoteWeight(u) = \frac{d(\vec{u}, \overline{c_u})}{\sum_{v \in C_u} d(\vec{v}, \overline{c_u})} \times |C_u| \tag{3}$$

其中，$\overline{c_u}$ 为簇 C_u 的质心所对应的向量，$d(\vec{v}, \overline{c_u})$ 表示词向量空间中向量 $\vec{v}$ 到向量 $\overline{c_u}$ 的欧氏距离，$|C_u|$ 表示簇 C_u 所包含的词语数量。公式(3)表明，一个簇的总投票分值为簇所包含的节点数量，簇内每个节点的投票权重根据距离质心的欧氏距离按比例分配，距离质心越远则投票的重要性越高。

当把两个节点在词向量空间中的语义关联关系表示为节点之间的聚类加权影响力，在进行聚类分析并计算得到每个词语的投票重要性后，笔者提出公式(4)计算节点之间的聚类影响力转移概率。

$$p_{cluster}(u \to v) = \frac{VoteWeight(v)}{\sum_{w \in adj[u]} VoteWeight(w)} \tag{4}$$

3.3 转移矩阵计算与关键词抽取

根据链接分析理论，只要给定图中节点之间的跳转概率转移矩阵，节点重要性就可以通过迭代计算得到。令矩阵 M 表示词图节点之间的概率转移矩阵[2]，如公式(5)所示。

$$M = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \tag{5}$$

其中，M 中的第 j 列表示由词语节点 j 跳转到其他节点的概率分布，每列的跳转概率之和为 1。相应的，p_{uv} 表示由节点 u 跳转到节点 v 的转移概率，即 $p_{uv} = p(u \to v)$。

为改进 TextRank 关键词抽取效果，笔者在先前提出的词语位置加权基础上[2]，进一步融合词向量聚类加权，对随机跳转概率 $p(u \to v)$ 进行合理赋值。

在文献[2]中，令 $p_{cov}(u \to v)$ 表示 $u \to v$ 的覆盖影响力，通过公式(2)计算求值，代表传统的 TextRank 的投票贡献。令 $p_{loc}(u \to v)$ 表示 $u \to v$ 的位置影响力，通过公式(6)计算[2]。

$$p_{loc}(u \to v) = \frac{I(v)}{\sum_{v_k : adj[u]} I(v_k)} \tag{6}$$

其中，$I(v)$ 表示节点 v 的位置重要性，参照文献[2]的实验结果，当 v 出现在标题中时，令 $I(v) = 30$，否则 $I(v) = 1$。

根据节点之间的覆盖影响力、位置影响力和聚类加权影响力，笔者提出公式(7)来计算节点 $u \to v$ 之间的跳转概率。

$$p(u \to v) = \alpha \times p_{cov}(u \to v) + \beta \times p_{loc}(u \to v) + \gamma \times p_{cluster}(u \to v) \tag{7}$$

其中，$\alpha + \beta + \gamma = 1$。利用公式(7)即可生成最终的转移矩阵 M。对于以候选关键词表示的文档 $d = \{w_1, w_2, \cdots, w_n\}$，参照文献[2]的处理策略，设定词图节点的初始分值如公式(8)所示。

$$B_0 = \left(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}\right) \tag{8}$$

则进一步采用公式(9)进行迭代运算[2]。

$$B_i = d \times M \times B_{i-1} + (1-d) \times \frac{e}{n} \tag{9}$$

其中, e 是一个所有分量均为 1 的 n 维向量, B_i 表示第 i 次迭代运算结束后, 每个词图节点的分值。当两次迭代运算结果 B_i 与 B_{i-1} 之间的差异非常小, 趋近于 0 时, 停止迭代, 此时, 每个节点的得分即为其在图中的重要性程度, 按照其得分大小降序排序, 挑选前 TopN 个节点作为关键词抽取结果, 实现关键词抽取。

4 实　验

4.1 实验数据

选取 2015 年 6 月发布的维基百科中文导出数据 "zhwiki-20150602-pages-articles-multistream.xml.bz"。该数据集共包含 2 648 029 个页面, 其中文章页面共有 1 480 963 个, 占页面总数量的 55.93%。通过过滤跳转文章和内容较短的文章等数据清洗处理之后, 最终保留了 516 695 个页面。通过中文分词工具 Ansj 进行分词[9], 形成供 Word2Vec 学习的文本数据集, 然后采用 Gensim 的 Word2Vec 模块[10], 以默认参数(采用 CBOW 模型、维度为 100、窗口大小为 5)对这批文本数据进行训练得到词向量模型文件①。

针对关键词抽取测试数据集, 文献[2]利用正文自动抽取算法, 提取 1 000 篇新闻报道的标题、正文和 META 字段中的关键词形成公开测试数据集, 但该数据集的关键词标注质量不高, 存在以标题本身作为关键词和关键词与内容相关度不高的情况, 因此, 本实验定向采集了南方周末网站的 1 524 篇文章, 提取其标题和正文, 并以网页中明确标记的标签作为文章对应的关键词, 构建形成新的测试数据集②, 该数据集中平均每篇文档包含 2 629.101 个字符和 3.565 个关键词。

4.2 实验结果及分析

为便于与已有方法对比分析, 笔者采用准确率 P、召回率 R 以及宏平均 F 值作为关键词抽取效果的评判标准, 令 K_A 表示测试数据集中文章本身所提供的关键词集合, K_B 表示算法抽取出的关键词集合, 则 P、R 和 F 值的计算方法如公式(10)所示[2]。

$$P = \frac{|K_A \cap K_B|}{|K_B|}, \quad R = \frac{|K_A \cap K_B|}{|K_A|}, \quad F = \frac{2 \times P \times R}{P + R} \tag{10}$$

与文献[6-7]保持一致, 实验中首先提取 3、5、7、10 个关键词作为自动抽取所保留的关键词与数据集本身提供的关键词进行对比。实验中对比的方法有:

M1: 文献[1]提出的最初的 TextRank 关键词抽取。

M2: 文献[6]提出的基于 Word2Vec 的词向量聚类关键词抽取。

M3: 文献[7]提出的融合 Word2Vec 与 TextRank 的关键词抽取。

M4: 文献[2]提出的词语位置加权 TextRank 关键词抽取。

M5: 本文提出的词向量聚类加权 TextRank 关键词抽取。

其中, 实验中所涉及的聚类分析部分均采用 K 均值聚类法, 聚类迭代次数为 20, 对比方法涉及的其他参数取各自文献中采用的最优值。同时, 笔者公开了所有相关数据和代码, 以方便读者对比或重现实验③。实验结果如表 1 所示。

表 1　不同关键词抽取算法的结果对比(TopN 取 3,5,7,10)

	TopN = 3			TopN = 5			TopN = 7			TopN = 10		
	P	R	F	P	R	F	P	R	F	P	R	F
M1	0.304	0.259	0.277	0.230	0.326	0.267	0.188	0.372	0.247	0.151	0.424	0.221
M2	0.119	0.191	0.143	0.095	0.240	0.131	0.080	0.263	0.116	0.072	0.295	0.107
M3	0.019	0.016	0.017	0.017	0.024	0.020	0.016	0.032	0.021	0.018	0.051	0.027
M4	0.356	0.306	0.326	0.270	0.383	0.313	0.217	0.428	0.284	0.170	0.479	0.249
M5	0.369	0.316	0.337	0.276	0.391	0.320	0.218	0.430	0.286	0.169	0.477	0.247

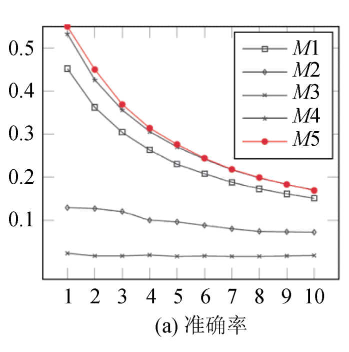
①实验生成的维基百科文本数据集和词向量模型文件, 可访问以下网址获取: https://github.com/iamxiatian/ x-extractor/.

②测试数据集保存成为 XML 格式, 可以从以下网站下载: https://github.com/iamxiatian/x-extractor/tree/master/data/articles.xml.

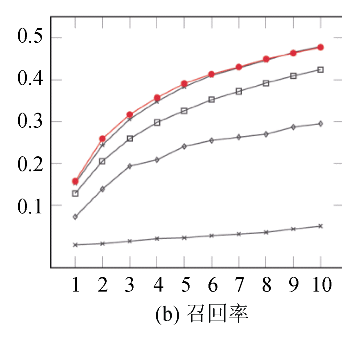
③https://github.com/iamxiatian/x-extractor/.

由表 1 得出，对于单文档关键词抽取，基于文章结构信息和投票机制的 TextRank 方法显著优于词向量聚类法(方法 M2)，词向量聚类法虽然能够把语义相关联的词语聚到一起，但选择距离质心最近的词语作为关键词效果并不理想；与之前研究结论相吻合，对 TextRank 的跳转概率做非均匀加权能够改善关键词抽取效果。然而，方法 M3 在节点重要性传递时，直接计算节点之间的词向量余弦相似度，以更大概率转移到相似度更高的词语上，在测试数据集上的表现效果较差。与方法 M3 不同，本文首先通过聚类确定词语在词向量空间下的语义簇，在簇范围内根据节点与质心的距离确定其投票重要性，当保留的关键词数量小于等于 7 个时，均优于其他方法，表明词向量聚类加权能够提升重要关键词的排序结果。

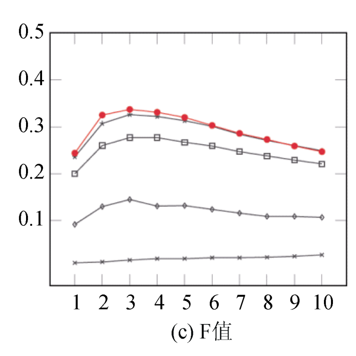
为全面观察不同关键词抽取方法的差异，笔者以曲线形式进一步给出了 TopN 取值在[1,10]时，5 种方法的准确率、召回率和 F 值的整体变化情况，如图 2 所示。整体看来，词向量聚类加权 TextRank 方法和词语位置加权 TextRank 方法，显著优于所对比的其他三种方法；当 TopN 取值限定在 5 个及以内时，词向量聚类加权比词语位置加权有较为明显的改进效果，当 TopN 取值不断增大，词向量聚类加权与词语位置加权的抽取效果差异不大。当 TopN = 3 时，方法 M4 和 M5 的 F 值同时取得最大值，此时，M5 比 M4 增量提升了 3.374%。



(a) 准确率



(b) 召回率



(c) F值

图 2　TopN 取值[1, 10]时，方法 M1 至方法 M5 的准确率、召回率和 F 值

为观察本文方法在抽取效果较差的情况下具体的关键词输出结果，笔者挑选与标注关键词结果完全不同的文档，把各种方法的输出结果组织到表 2 之中，其中，各个方法抽取结果保留的关键词数量与原始文档给出的标签数量保持一致①。

表 2　完全未命中原始关键词的抽取结果示例

文档编号	101037	24576	26808
标注结果	民企，军工，融合	日本侵华，轮船，索赔，陈春	财政部，金融高管，限薪
M1	政府，公司，日本	日本，陈顺通，陈洽群，律师	薪酬，金融机构，国有
M2	企业，政府，日本	陈顺通，幼子，上海，三井	国有，金融机构，水平
M3	企业，政府，日本	租金，见证，航运业，预定	征求，相关，监事长
M4	军火，企业，日本	船王，日本，陈顺通，陈洽群	金融机构，薪酬，国有
M5	军火，企业，政府	船王，民间，日本，陈顺通	金融机构，国有，薪酬

对于表 2 所示的抽取结果完全未命中的情况，方法 M1、方法 M4 和方法 M5 均属于词图迭代计算方法，输出结果具有较高的重叠性，能够在一定程度上代表文档的主要内容，M2 的部分结果与文本有一定关联，而 M3 的结果相对较差。综上分析，可以得出结论：

(1) 对于单文档直接应用词向量聚类分析，选择每个聚类簇的代表性词语作为关键词，效果不佳。

(2) TextRank 在单文档关键词抽取方面具有较为稳定的效果，通过词语位置加权和词向量聚类加权可以进一步提升 TextRank 的抽取准确性。

5　结　语

本文以词向量聚类加权方式，将维基百科的世界知识纳入到 TextRank 的关键词抽取过程中，以改善关键词抽取效果。与基于逆文档频率或 LDA 的改进方法不同，词向量的训练不依赖于关键词所在的数据集，

①字符串“http://www.infzm.com/content/”后面附加上文档编号，即为文档的 URL 访问地址。

抽取结果相对更为客观稳定。实验结果表明，保留的关键词数量越少，词向量聚类加权的抽取效果改善越显著，当保留的关键词数量 TopN 超过 7 以后，聚类加权与单纯的位置加权没有显著差异。

下一步的研究内容包括：探索更合理的词向量聚类结果加权方法；从序的角度对关键词抽取结果进行全面评价。

参考文献：

[1] Mihalcea R, Tarau P. Textrank: Bringing Order into Texts [C]//Proceedings of Empirical Methods in Natural Language Processing. 2004.

[2] 夏天. 词语位置加权 TextRank 的关键词抽取研究[J]. 现代图书情报技术，2013 (9): 30-34. (Xia Tian. Study on Keyword Extraction Using Word Position Weighted TextRank [J]. New Technology of Library and Information Service, 2013 (9): 30-34.)

[3] 顾益军，夏天. 融合 LDA 与 TextRank 的关键词抽取研究[J]. 现代图书情报技术，2014 (7/8): 41-47. (Gu Yijun, Xia Tian. Study on Keyword Extraction with LDA and TextRank Combination[J]. New Technology of Library and Information Service, 2014(7/8): 41-47.)

[4] 李鹏，王斌，石志伟，等. Tag-TextRank: 一种基于 Tag 的网页关键词抽取方法[J]. 计算机研究与发展，2012, 49(11): 2344-2351. (Li Peng, Wang Bin, Shi Zhiwei, et al. Tag-TextRank: A Webpage Keyword Extraction Method Based on Tags [J]. Journal of Computer Research and Development, 2012, 49(11): 2344-2351.)

[5] 谢玮，沈一，马永征. 基于图计算的论文审稿自动推荐系统[J]. 计算机应用研究，2016, 33(3): 798-801. (Xie Wei, Shen Yi, Ma Yongzheng. Recommendation System for Paper Reviewing Based on Graph Computing [J]. Application Research of Computers, 2016, 33(3): 798-801.)

[6] 李跃鹏，金翠，及俊川. 基于 Word2vec 的关键词提取算法[J]. 科研信息化技术与应用，2015, 6(4): 54-59. (Li Yuepeng, Jin Cui, Ji Junchuan. A Keyword Extraction Algorithm Based on Word2vec [J]. e-Science Technology & Application, 2015,6 (4): 54-59.)

[7] 宁建飞，刘降珍. 融合 Word2vec 与 TextRank 的关键词抽取研究[J]. 现代图书情报技术，2016 (6): 20-27. (Ning Jianfei, Liu Jiangzhen. Using Word2vec with TextRank to Extract Keywords [J]. New Technology of Library and Information Service, 2016(6): 20-27.)

[8] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [C]//Proceedings of Workshop at International Conference on Learning Representations. 2013.

[9] Ansj Lexical Parser [EB/OL]. [2016-10-01]. https://github.com/NLPchina/ansj_seg.

[10] Deep Learning with Word2vec [EB/OL]. [2016-10-01]. http://radimrehurek.com/gensim/models/word2vec.html.

利益冲突声明：

作者声明不存在利益冲突关系。

支撑数据：

支撑数据见 https://github.com/iamxiatian/x-extractor/tree/master/data/articles.xml。

[1] 夏天. articles.xml. 关键词抽取的测试文章集合.

收稿日期：2016-10-28
收修改稿日期：2016-12-16

Extracting Keywords with Modified TextRank Model

Xia Tian
(Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education,
Renmin University of China, Beijing 100872, China)
(School of Information Resource Management, Renmin University of China, Beijing 100872, China)

Abstract: [**Objective**] This study aims to improve the single document keyword extraction algorithm by adding the world knowledge vector from the Wikipedia to the TextRank model. [**Methods**] First, we created a new word embedding model based on the Word2Vec model with Wikipedia's Chinese data. Second, we clustered the nodes of TextRank wordgraph to adjust the voting importance of each cluster. Third, we calculated the random walk probability with additional factors of coverage and location. Finally, we got the node score with iterative computation of the transition matrix, and then selected the Top N words as the needed keywords. [**Results**] The performance of the new TextRank model was much better than other methods when the Top N value was less than or equal to 7. If we only retrieved three keywords, the F measure reached its maximum value, which was 3.374% higher than the best existing results. When the Top N value was larger than 7, the results were similar to the traditional TextRank method. [**Limitations**] The computation cost was increased due to the cluster analysis. [**Conclusions**] The new weighted TextRank model could extract keywords effectively.

Keywords: Keyword Extraction　Word Embedding　TextRank　Word2vec

Knowledge Unlatched 和 JSTOR 合作研究如何利用开放获取图书

人文社会科学专著开放获取支持计划 Knowledge Unlatched(KU)和 JSTOR 数字图书馆正在合作研究开放获取资源的使用模式。虽然 KU 将继续在 OAPEN 和 HathiTrust 平台上托管资源, 但同时也会将 30 多个开放获取图书资源交由 JSTOR 托管, 包括历史、文学、政治科学、人类学和媒体与传播等领域的图书, 所有这些都是由世界领先的学术出版社出版的, 并在世界各地的图书馆的支持下成功"解锁"(开放获取)。

KU 总经理 Sven Fund 博士说: "内容的广泛使用是开放获取的核心, 这对 Knowledge Unlatched 来说是非常重要的。在我们的倡议下出版商和图书馆一直在共同努力寻求资源访问的便捷性, 从而扩大人文社会科学专著的使用。"

JSTOR 图书馆馆长 Frank Smith 表示: "我们很高兴与 Knowledge Unlatched 合作, 为全世界的研究人员提供这些高质量的图书。我们期待双方能深入研究, 以了解开放获取图书的使用情况和影响力。"

(编译自: http://www.knowledgeunlatched.org/2017/02/jstor/)

(本刊讯)